How to cite curated databases and how to make them citable

Buneman, P. (2006). How to cite curated databases and how to make them citable. In Proceedings of the 18th Conference on Scientific and Statistical Database Management. (pp. 195-203). IEEE Computer Society. 10.1109/SSDBM.2006.28

How to cite curated databases and how to make them citable

Peter Buneman
University of Edinburgh

Professor Tony Harmar
School of Biomedical Sciences
University of Edinburgh

Dear Tony,

Please forgive this rather lengthy discussion of citation. This letter started life as a short e-mail follow-up to our discussions on the use of persistent object identifiers as citations, but after talking to our colleagues,[1] *a whole collection of closely related issues emerged concerning citation in databases. I had thought that finding a citation scheme for the IUPHAR [12] database would be straightforward, and in some sense it is; but after scouring the internet, I could find no help on the topic. While a number of organisations stress the importance of citing databases, it appears that no one has seriously considered the issues involved in citing all or parts of a something that has internal structure and that evolves over time. The point of writing to you at length is partly to understand the role of persistent object identifiers in citation, but more importantly to understand how one should cite a part of a database, and how one makes the database citable.*

What I want to propose is a stable citation system for IUPHAR which should also work for a wide variety of other curated databases. In particular, I want to describe how to publish the database in a form that can be cited, how to ensure that the citations remain valid and how to generate and validate the citations automatically

All of these require a little extra work, but I believe we have enough technology in place to make this possible. Please let me know what you think.

With best wishes
Peter

This paper appeared in the Proceedings of the 18th International Conference on Scientific and Statistical Database Management, Vienna, July 2006. Some small corrections have been made in this version.

1 Preliminaries

Curated scientific databases such as the IUPHAR database resemble conventional publications such as reference manuals in that they represent the work of a large number of people who both create and revise their contents. The difference is that curated databases have more internal structure and that they change more frequently. How should we cite all or parts of such a database? We use conventional citations primarily to identify the source material, but this is not their only use. They are distinguished from persistent object identifiers (or other "randomly" assigned digital keys) in their ability to provide some additional information, such as authorship or title, that may be useful even before we look at the cited work. As mechanisms for identification they are usually highly redundant. For example, Bard JB and Davies JA. Development, Databases and the Internet. Bioessays. 1995 Nov;17(11):999-1001. is much more than we need to identify the work. Bioessays 17:999-1001 is sufficient, so, almost certainly, is the combination of authorship and title. The citations Ann. Phys., Lpz 18 639-641 and Nature, 171,737-738, while adequate for identification, hardly convey their well-known identities.

We should note that persistent object identifiers [7, 1] are not just identifiers; they have supporting mechanisms for retrieving the associated "digital object". By contrast, a citation does *not* give us a specific mechanism for retrieving a document. It is a structure that can be used by a variety of mechanisms such as online indexes and search engines; it is also useful (when, once we have found the containing document such as the journal or issue) to find what we are looking for. In fact, a citation consists of two kinds of information which, for want of better terms, I shall call *location* information such as Bioessays 17(11):999-1001 and *descriptive* information such as authorship, title, date. This distinction will be especially important for databases, which have an internal structure that is richer and different from that of documents. We should also note that the descriptive information is to some extent arbitrary. There is no canonical citation, and two textually distinct citations may identify the same thing.

What kind of citation will provide the location and descriptive information for some part of a database? Let

me start by stating some requirements concerning citations that I believe are obvious to anyone working in traditional scholarship: there is some "thing" that is being cited; the thing should be accessible; and the thing should not change over time. Despite the fact that database technology is now in widespread use for scientific publishing, there are few accepted practices for supporting citation of data: there are few standards, there is little supporting technology, and the requirements above, if they are met at all, are met in an *ad hoc* fashion.

For brevity, I want to make use of a small amount of notation. If C is a citation then $\langle C \rangle$ is the thing being cited. For example if the citation is Life Sci., 53, 393 - 398, then $\langle$Life Sci., 53, 393 - 398$\rangle$ is the article being cited.

The first of a series of desiderata that I propose for databases arises immediately from the requirements above:

D1 *For any citation C, $\langle C \rangle$ should remain fixed*

Since databases change, this simple requirement is not always easy to maintain; we shall return to it later. The second is that anything we cite should provide us with at least one way of citing it:

D2 *Any citable thing T should contain a citation C such that $\langle C \rangle = T$*

This is not always done in journal publications (presumably because the citation can be figured out from the enclosing issue of the journal.) It is essential, I believe, for electronic publications. The reasons for requiring it in web pages are almost obvious. First, one wants confirmation that we have found the correct citation. Even if we found T using some other citation C' (that is $\langle C \rangle = \langle C' \rangle$), we would expect there to be sufficient commonality between C and C' to be sure that they refer to the same thing. In particular, we expect the location information to agree. Second, if we found $\langle C \rangle$ by some other means, such as a search engine or by finding a copy somewhere, we would want to know how to cite it. Finally, it may be that one wants the citation to carry some important descriptive information, such as authorship, which may not be necessary for identification, but is desirable in the "authoritative" citation.

2 Current Practice

On-line databases frequently give recommendations on how to cite them, but these are seldom satisfactory. They often omit version information or fail to provide adequate location. There is also a fair amount of literature on how to cite on-line data, but it is apparent from looking through this that databases are problematic. The *Columbia Guide to Online Style* [17], although it discusses issues of permanence of links, does not mention D1 as one of its citation "principles". There is a section of the ISO690 standard [11] (itself difficult to cite!) that deals with citations of parts of electronic documents. Another report [15] goes into some detail on how to cite databases and parts of databases. It suggests, as an example,

> Nutrition Education for Diverse Audiences [Internet]. Urbana (IL): University of Illinois Cooperative Extension Service, Illinet Department; [updated 2000 Nov 28; cited 2001 Apr 25]. Diabetes mellitus EFNEP lesson; [about 1 screen]. Available from: `http://www.aces.uiuc.edu/~necd\-/inter2\_search.cgi?ind=854148396`

The usefulness of the location information in this is questionable: the http parameter ind=854148396 is likely to depend on the session, and whether you are 1 screen or 3 screens into the data will surely depend on the configuration of your browser.

It would be easy to continue to find fault with such recommendations, but the truth of the matter is that the writers of these manuals are doing the best they can with what is "out there". The fault lies with the database curators who have failed to provide a stable citation system for their databases and the computer scientists who have failed to provide the supporting technology. In what follows I want to suggest how to redress the situation.

3 Structural issues

We need first to understand location information and the degree to which a citation enables one to localise the relevant material. A complaint I have heard from curators who check the validity of citations is that they spend an inordinate amount of time searching the cited text. For example, suppose the citing text reads "In C it is claimed that P". If P is a direct quote, we may be able to search for it efficiently in an on-line article. But if the article is paper, or if P is not a direct quote, it may be time-consuming to locate the relevant text.

Databases are distinguished from traditional publications by the degree of explicit structure. This offers the possibility of a citation using this structure to home in on the relevant data. To understand the possibilities, let us use the IUPHAR database as an example. The structure of the web pages as they appear through the web interface is shown in Figure 1, in which the arrows represent hyperlinks. It is a testimony to the organisation of your data and its presentation that a non-

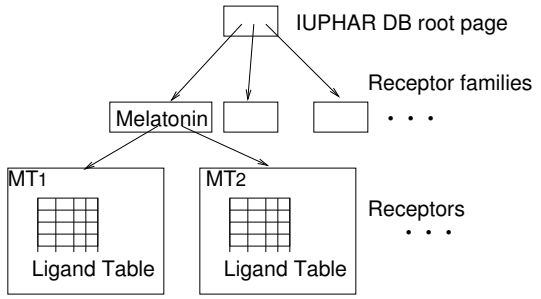

Figure 1. Rough structure of the IUPHAR web interface

biologist like me can make some sense of what is going on. This kind of organisation is common in curated biological databases (e.g., [14, 9]); and in scholarship generally. Gazetteers (e.g., [5]), dictionaries and other curated reference materials present a similar structure.

Let us make a temporary assumption that the database is fixed – there is only one "version" of it.

My understanding of the structure of the IUPHAR database *as it is seen by someone browsing the interface* is that the major component is a list of receptor families; for each family there is a list of receptors; for each receptor there is a web page where the main technicalities appear. This web page has substantial internal structure, such as a table of ligands and their function for that receptor. Note that the structure of what the user sees is not the same as the underlying database. In the case of IUPHAR, the underlying database is relational, and the web pages show a hierarchical structure that is generated by your software. Again this is common practice. In what follows, when I refer to the "database" I shall mean the structure perceived by someone browsing the web interface. I shall use the term "underlying database" for the (relational) database from which the web interface is generated.

Consider the following fanciful references of the IUPHAR database, where C_1, C_2, C_3 are citations in the text:

1. The IUPHAR database (C_1) contains no information about Ginandtonicin.
2. The IUPHAR database (C_2) lists five ligands for Melatonin receptor MT_1.
3. The IUPHAR database (C_3) asserts that luzindole is an antagonist ligand for receptor MT_1.

For claim 1 C_1 should refer to the whole database. For 2 it would be appropriate for $\langle C_2 \rangle$ to be the web page for that receptor or maybe the receptor family page. Claim 3 is attested in a row of a tabular display that appears in a receptor web page. One could imagine citing just that row or the table. It is more likely, though, that one would cite the receptor or its family. Because of small size and the well laid-out structure of the web pages, it is an easy matter to verify that the row actually occurs in the receptor web page. In the case of 3 the row of the table alone does not identify the relevant receptor; that information occurs in the enclosing web page, so citing the row alone will probably not tell us what we want to know. Making the context too narrow can be as counterproductive as making it too wide. Let us assume that, following Figure 1, the presentation of the database is hierarchical and say that one citation is *coarser* than another if it refers to a higher structure. In the example above $\langle C_1 \rangle$ is coarser than $\langle C_2 \rangle$. This brings us to another *desideratum* of database citation.

D3 *It should be possible to cite a database at varying degrees of coarseness.*

This does not mean that we need to cite a database at all levels of coarseness; rather that the citation system should allow more than one level if needed. For example, one can imagine citations of the whole database and of receptor families both being useful.

In order to make further progress, we now have to look at the internal structure of a citation. When we see a citation like Life Sci., 53, 393-398, we understand from the order and format of the components that the journal is Life Sci., the volume number is 53 etc. Our understanding is based on a common structure of all journals. When it comes to databases we have to be explicit about the structure. So, if we are talking about a receptor-family in IUPHAR, we need to be explicit about this in the citation.

It will help to adopt what, in the jargon of computer science, we call a "concrete syntax" for citations, which is a sequence$\{k_1{=}v_1, k_2{=}v_2, \ldots\}$ where $k_1, k_2, \ldots$ are keywords and $v_1, v_2, \ldots$ are associated values. For example, {Journal="Life Sci.", Number=53, Pages=393-398}. We could equally well use one of a number of other formats such as a format that separates the location and descriptive information. Of course, what is important is the *abstract* syntax, the keywords and the information conveyed by the associated data. The Dublin Core Metadata [8] is an example of an abstract syntax for bibliographic data.

Given such a structure, there is a natural "part-of" relationship among citations. For example, {Journal= "Life Sci."} and {Journal="Life Sci.", Number=53} are both meaningful parts of the citation above. There is

no implication that all parts of a citation are meaningful on their own: the citation {Number=53} is unlikely to be of much use. If we look at a possible citation structure for receptor families in IUPHAR, the one that naturally presents itself is the form {DB=IUPHAR, Family=Melatonin}. Here {DB=IUPHAR} is a meaningful coarser citation, while {Family=Melatonin} is not. Now, one could imagine an alternative citation system in which each receptor family is independently citable, e.g. {IUPHAR-Receptor-family=Melatonin}. I believe it is still useful to keep a reference to the coarser database, bringing up the next desideratum:

D4 *If C and C' are citations and $\langle C' \rangle$ is coarser than $\langle C \rangle$ then the location information in C' should be part of the location information in C*

Even if {IUPHAR-Receptor-family=Melatonin} is adequate to identify the relevant page, it is better to use {DB=IUPHAR, IUPHAR-Receptor-family=Melatonin} as the full location information. This is probably the most contentious requirement. Arguably, if we can find $\langle$ {IUPHAR-Receptor-family=Melatonin} $\rangle$ and if that page contains an "up" link to the coarser page, there is no need for the coarser citation. However, there are too may "if"s, and when we come to look at versions there are more compelling reasons for wanting this.[2]

4 Temporal issues

Now let us address the fact that databases change. This complicates the process both of preservation and citation. Before going into how this affects citation, it is worth looking at the nature of the change. The first and obvious kind of change is the addition of new material to an existing data set, maybe a new receptor or ligand. This kind of change is to be expected in scholarship, but what about *modification* – the change in which existing data elements are overwritten? This can happen for a variety of reasons. I am sure that there are cases in the IUPHAR database in which corrections are made. There is very little in this database that is "raw" data. Much of it is judgements made on the basis of existing experimental evidence, and this inevitably gets revised. Another source of change occurs when the object of study itself changes. This is less likely to be an issue in your field, but it is certainly a major issue in, for example, gazetteers where demographic, political and economic information is constantly changing.

The obvious way to deal with change in citation is to provide, in the citation, a version number, for example {DB=IUPHAR, Version=17, Family=Melatonin}; but this immediately raises two questions: why not use time rather than a version number, and what does the version refer to (in this case, the database or the family?) First, I want to argue that using time may be misleading. I have been using time in the citations in this for this note because I could not find anything better, but this is the time at which I *retrieved* the material, not the time at which it was created. There is no global synchronisation on the internet so if two people give out identical citations of this form, there is no guarantee that they are citing same thing. Of course, we could use version creation time as the identifier or as a part of it, but this might make it difficult to find, from the citation, next or previous versions of the database. Surely we should adopt the practice of conventional citations and include the time (e.g. the year and month) as useful descriptive information. Biological databases vary widely in how frequently new versions are "released". In the case of Uniprot/Swissprot [9] the period is months whereas for OMIM [14] the period is, or was, hours or days.

Second, to what does the version refer? It could be the receptor, the receptor family, the database, or – going beyond this – some collection of databases or the whole web. The last of these is clearly nonsensical: there is no way we can talk about the state of the web at a given instant. What distinguishes a database from any larger structure is that of *integrity*. Within a database certain constraints are enforced, quite often by the database management system itself. For example, that there are no "dangling pointers" within your database is probably enforced by the underlying database management system. There are no such guarantees on references to material outside your database. For our purposes, the defining characteristic of a database is that it is the coarsest level at which integrity or internal consistency is maintained. With this:

D5 *Versions should be recorded at the database level*

This may seem unintuitive. Every time one changes, say, a receptor page, one creates a new version of the database. This is annoying, perhaps, for someone interested in another receptor to see that the version has changed even though the data for that receptor has remained unchanged. Consider the alternative: someone citing the whole database, perhaps because they have performed a query that involves the whole database, will have to cite the versions of each individual receptor that the query looked at. Worse, such a query is hardly meaningful. There is no apparent guarantee that the version of the database did not change while the query was in progress. In practice, the rate of *publication* of versions is much slower than the rate of updates. You publish new versions of the database relatively infrequently; and this policy appears to be common in curated databases such as yours. It is therefore unlikely that you will want very large version numbers.

There is no harm in in large version numbers and they can be turned into compounds, such as {... Edition=5, Version=42...} in which both edition and version are needed to specify the state of the database, but changes in edition are associated with larger, perhaps structural, changes to the database.

Our conclusion so far is that a correct citation of some part of the database will now contain some indicator of both a location in the hierarchical structure of the database and a version, for example, {DB=IUPHAR, Version=17, Family=Melatonin}. Having such a citation obliges you, or someone, to keep past versions, so that ⟨{DB=IUPHAR, Version=17, Family=Melatonin}⟩ can be found.

An important observation on versions is that one may want to cite a database over a certain period. Such citations against the IUPHAR database are a bit contrived, e.g. "The number of receptor families catalogued in IUPHAR {...} has been steadily rising". However, in databases in which there is an important historical record, such citations may be particularly important, e.g. "Over the last 10 years {...}, the GDP of Lichtenstein rose by an average of...". In such cases it is possible to cite a range of versions, such as {... Version=12-21, ...}[3]. Temporal queries on such databases are discussed in detail in [16].

Now, what is ⟨{DB=IUPHAR, Family=Melatonin}⟩, a citation without a version number? The answer we probably want is that this is the latest version of the database. This means that, while {DB=IUPHAR, Family=Melatonin} is a perfectly useful construct in that ⟨{DB=IUPHAR, Family=Melatonin}⟩ exists and is useful, it is not good practice to use it as a citation, because it changes (violating D1). In web terminology we probably need two words: one for a fixed citation and one for a "current link" – the place at which you may find the latest information.[4] In this context, some XML committees (e.g., [18]) do a good job of distinguishing between "this" version, the "latest" version and previous versions of documents.

5 Descriptive information

There is little more to be said about descriptive information in citations to databases other than that it is likely to be different than what we use in conventional citations. For example, in IUPHAR, I note that you use the term "contributors" for the people who work on a particular receptor family. A title is not needed because the receptor name is used in the location information. In the case of a database, the time of last update of the cited part is often useful to convey the currency of the data. Thus, {DB=IUPHAR, Version=17, Family=Calcitonin, Contributors="D. Hay, D.R. Poyner", Last-update = 10/10/2005} is a possible citation.

6 Presentation, content and preservation

Throughout the discussion so far we have assumed that what is being cited has some form of hierarchical structure, the structure that the user of the database sees when looking at the relevant web pages. This structure is not necessarily the same as the structure of the database from which those web pages have been constructed. This is certainly the case in the IUPHAR database. Moreover, the underlying database almost certainly contains information – such as working notes or data required to make the database perform efficiently – that is not intended as part of the published material. Clearly, we should not be making direct citations to the internal structure of the database.

On the other hand, should the cited "thing" be what the user sees on the screen? This is equally problematic, for even though you have done your best to produce a useful interface, you cannot be sure that the user's browser is functioning properly, nor do you have any guarantee that some other "screenscraper" has not taken the web pages that you export and re-organised or otherwise mangled the presentation. Even if one did have those guarantees, there are almost certainly details of the presentation, such as font size, page length, colours, browsing patterns etc. that are irrelevant. So the presentation, even if it were possible to give it a precise characterisation, is also not appropriate for citation. Moreover, the preservation of what the user sees (D1) may be problematic. We need guarantees that the browser etc. will not change and that you have preserved your web interfaces as well as your database.

So what should we regard as the cited thing? In general this is a problem with no clear answer, but in the case of a structure such as the one you present, there is a simple solution: the hierarchy that the user sees should be represented as an XML document. The users should be aware that they are seeing a display or rendering of parts of that document; they should be able to understand and to retrieve those parts (the parts that they cited) if needed. It appears from the structure of your web pages that this is a straightforward thing to do, and – if the database is at all complicated – there are tools for efficiently publishing relational databases as XML documents [2].

Nowadays there is justified concern about the long term preservation of digital materials. There are two issues here: first is simply preserving the bits [13]. It is sur-

{DB=IUPHAR, Version=$v, Family=$f} ← /Root[ ]/Version[Number=$'v]/Data[ ]/Family[FamilyName=$'f]

Figure 2. A rule that generates location information

prisingly difficult to obtain the same longevity as we get from ink and paper. The second is preserving the *interpretation* of those bits, which is the purpose of representation information [6]. For example, it would be considerably more difficult to preserve the current presentation of IUPHAR databases as web pages than it would be to preserve the corresponding XML document. The former requires you to preserve the software you wrote, browser, and maybe the underlying operating system. The latter is simply a text file.

Should one preserve any more representation information than XML file? Obviously some kind of schema and textual description is going to be helpful, but well-designed XML is eminently readable. A schema or some other representation information may be useful as an integrity check, but provided the XML itself contains descriptive tags and does not use numerical codes or other devices for compressing data, my prediction is that hundreds of years from now, a biologist will be able to understand a well-structured XML representation of the IUPHAR database, even without the schema. It will not require the genius of Champollion or Ventris to decipher it.

To summarise the discussion of presentation and preservation, I suggest that you publish your data as an internally versioned XML document. The software that we are currently developing for your system to archive the underlying database [4] is also designed to archive versions of XML documents efficiently. Also, as we observed earlier, persistent identifiers are no substitute for citations; however, they should be included in citations where appropriate.

7 Automatically generating citations

If we are generating an XML document as the citable structure, then – following D2 – that document should contain its citations in the appropriate locations. Each citable component of the document should have a subcomponent, perhaps labelled Citation, which tells us how to cite it. There should be sufficient information in the document to specify the contents of the citation, and the citation should be generated automatically. The most obvious reason for wanting this is that to insert citation data manually is both time-consuming and error-prone. But having such a system is also a good check on the *integrity* of the document: it can guarantee that the contents of the document are consistent with the citation. One would like to require that the information needed to create a citation for a node always exists and that it specifies preciseley that node. One may also want guarantees on the descriptive information, e.g. that a given node has at most one Title or that it has exactly one DOI (digital object identifier).

If you have read this far, you will be aware that I have been relegating the computer science technicalities to endnotes such as this[5], but I now want to expose some examples of citation specification in order to show that it is simple and in order to describe the kinds of constraints it places on your published data. Figure 2 shows an example of a citation specification that produces only location information.

The expression to the left of the arrow is in our concrete syntax of citations with variables such as $\$v$ and $\$f$. When particular values are substituted for these variables we get a citation such as {DB=IUPHAR, Version=17, Family=Melatonin}. The stuff to the right of the arrow is a *pattern* which is expected both to match the node being cited and to provide values for the variables. The pattern is expressed in the syntax of XPath, a language for specifying sets of nodes in an XML document. Here, however, we are using it to *constrain* the XML document and to provide values for the variables. It is worth describing how these constraints work, because they have some impact on how you export your citable data. The pattern consists of a series of *steps* each started by a "/"

- The /Root[ ] step expresses the fact that the database or document has a unique root,[6] the top of the hierarchy.
- The /Version[Number=$'v] step says that under the root, we will find a number of Version nodes. Each Version must have a Number that uniquely identifies the node and provides a value for $\$v$.
- The /Data[ ] step indicates that for each Version, there is precisely one data node. (This data node contains the whole of the exported IUPHAR data for this version)
- The /Family[FamilyName=$'f] step specifies that for each data node there is a set of Family nodes, each of which must have a FamilyName which uniquely identifies the family.

{ DB=IUPHAR, Version=$v, Family=$f Receptor=$r, Contributors=$a, Editor=$e, Date=$d, DOI=$i}
  ←
/Root[ ]/Version[Number=$'v,Editor=$?e, DOI=$.i, Date=$.d] /Data[ ]/Family[FamilyName=$'f]
        /Contributor-list/Contributor=$$^+$$a] /Receptor[ReceptorName=$'r]

{ DB=IUPHAR, Version=11, Family=Calcitonin, Receptor=CALCR, Contributors={Debbie Hay, David R. Poyner},
  Editor=Tony Harmar, Date=Jan, 2006, DOI=10.1234}

Figure 3. A rule that generates description information and an example of what it generates

I hope these appear as obvious and reasonable constraints on any hierarchical structure which could be used to publish the IUPHAR data. Now let us look as an example of a specification that generates both location and descriptive information. Figure 3 shows such a rule and an example of a citation it could generate.

In the pattern in Figure 3, the step /Version[...DOI=$.i...] indicates that the DOI is associated with the version, which is, I believe, the appropriate referent or target for the DOI. If it is preferable to have a DOI for each family (of each version) then the appropriate place to place those identifiers is in the /Family[...] step. It is perfectly possible to have DOI at both levels, in which case they would have to be given different names in the citation Version-DOI and Family-DOI.

The variables in the pattern are decorated in ways that indicate the various further constraints we are placing on the document[7]. For example, $.d in the step /Version[...DOI=$.i...] indicates that exactly one value of the DOI is expected. The $?e indicates that at most one editor can exist, and the $$^+$$a in the Family[...] step indicates that one or more contributors are expected, in which case $a is a *list* of values.

Specifying constraints and generating citations could also be done in some combination of XML-Schema and XQuery. Such specifications would be quite impenetrable compared with what I have proposed here. Moreoever, constraint-checking mechanisms for XML-Schema may be expected to be much more complex [10].

8 Unresolved issues

There are a few points that need to be taken care of before "coding this up". I list some of them here, but I should emphasise that none of them have any serious impact on the general technique. They mostly concern the concrete syntax of what we generate for citations.

- If citations are also to be and machine-readable, shouldn't the concrete syntax be expressed in XML? Possibly, provided the XML can be kept human-readable.
- I have assumed that the key path in a citation specification pattern gets you to the node being cited. In the examples above, the two key paths are:
  Root[ ]/Version[...]/Data[ ]/Family[...]
  and
  Root[ ]/Version[...]/Data[ ]/Family[...]
  /Receptor[...]
  In the second case, we have to generate a citation for each receptor. But we could take the view that the citation resides at the Family level, and the /Receptor[...] step is just added descriptive information; i.e., some of the location information has become descriptive.
- There are some issues in the syntax of citations with sets or lists of values. Suppose we have {..., Contributors=$a,...} where $a is bound to a list of strings. One might want, for the purposes of formatting, to specify that a string-valued function to be applied to $a, e.g., Contributors=$f$($a$) where f creates a string with "and" between the last two contributor names, rather than ",". On the other hand, it is probably dangerous to apply such a function to location/key variables.

These points, taken together with the fact that we also need some standards for character sets and character strings, argue for the use of XML for concrete syntax and stylesheets to provide other formats. Until the community or communities decide on the basic standards, it is probably better to adopt a lightweight solution.

9 Conclusions

That's about it. The main point is that, in order to prepare databases such as yours for long-term accessibility and effective citation, we have to do a modest amount of work in structuring the data appropriately in XML, after which citations can be specified and generated by some simple rules. Moreover, the conformance of the XML document to the citation constraints can be

checked efficiently[8]. I believe it will not be hard to get this to work for the IUPHAR database.

There are, of course, a few unresolved issues with the scheme, and there is no doubt that whatever we do will eventually be "non-standard", but someone has to start somewhere, so why don't we do it?

Notes

[1]I am indebted to Jonathan Bard, Rajendra Bose, Carwyn Edwards, Wenfei Fan, Ann Matonis, Ed Rosser and Henry Thompson. I am especially grateful to Chris Rusbridge for his help with the existing literature on citation.

This work was supported by funding from the EPSRC (Digital Curation Centre) and from the Royal Society

[2]More formally, we can express the location information in a citation $\{l_1{=}v_1, \ldots, l_n{=}v_n\}$ as a conjunction of "atomic" citations, $\{l_1{=}v_1\} \wedge \ldots \wedge \{l_n{=}v_n\}$, with each $\{l_i{=}v_i\}$ expressing some property of the cited thing. The ordering on citations is implication. Assuming the cited structure is hierarchical, (we shall later suggest it is an XML document) an element T is *coarser* than an element T' ($T \geq T'$) if T is above (an ancestor of) T' in the hierarchy. The requirement D4 is that of monotonicity: if both $\langle C \rangle$ and $\langle C' \rangle$ exist then $C \Rightarrow C'$ iff $\langle C' \rangle \geq \langle C \rangle$.

[3] Computer scientists may again observe that the appropriate way to formalise $\{\mathsf{Version{=}12\text{-}18}\}$ is as a disjunction $\{\mathsf{Version{=}12}\} \vee \ldots \vee \{\mathsf{Version{=}18}\}$. The ordering is still implication, and a citation can be normalised into a disjunction of conjunctions. Then $\langle C_1 \vee \ldots \vee C_n \rangle$ is the set of elements $\{\langle C_1 \rangle \ldots \langle C_n \rangle\}$. We now have to "lift" the coarseness ordering on elements to an ordering on sets of elements. For this we use the ordering $\geq^S$ defined by $S_1 \geq^S S_2$ iff $\forall x_2 \in S_2 \exists x_1 \in S_1 . x_1 \geq x_2$. With respect to this ordering, $\langle . \rangle$ continues to be monotone.

[4]At first sight this destroys the monotonicity property; however, we could regard a citation C without a version number as the citation $C \wedge (\{\mathsf{Version{=}1}\} \vee \{\mathsf{Version{=}2}\}) \ldots$, i.e., a citation to all past present and future states of the database. With this interpretation the monotonicity property still holds, and the user of an "unversioned" citation is guilty of citing something that doesn't yet exist!

[5] Here are the details of the citation generation mechanism. The general structure is $C \leftarrow P$ where C is in the syntax of citations $\{a_1{=}\$x_1, \ldots, a_n{=}\$x_n\}$ augmented with variables $\$x_1, \ldots, \$x_n$. P is an XPath "pattern" shortly to be described. The idea is that P is matched at the node to be cited and will bind the variables $x_1, \ldots, x_n$.

To turn to the syntax of patterns, the starting point is XML keys [3] specified using the syntax of XPath. A *key pattern* is an XPath expression with decorated variables of the form:

$$\begin{array}{rcl} E & = & /t_1[p_1^1{=}\$'x_1, \ldots, p_1^{k_1}{=}\$'x_1^{k_1}]/\ldots \\ & & /t_n[p_n^1{=}\$'x_n^1, \ldots, p_n^{k_n}{=}\$'x_n^{k_n}] \end{array}$$

in which the t_i are tag names and the p_i^k are "fully specified" downward paths consisting of a sequence of tag names (no wildcards, no //). The pattern variables $\$x_1, \ldots, \$x_1^{k_1}, \ldots, \$x_n^1, \ldots, \$x_n^{k_n}$ are all distinct and contain the citation variables $\$x_1, \ldots, \$x_n$. We stress that E, although it exploits the syntax of XPath, and although we will formalise the constraints it imposes using the semantics of XPath, is not to interpreted as an XPath expression. It denotes a constraint and a binding mechanism for variables.

Using $[\![e]\!](c)$ for the set of nodes denoted by the XPath expression e acting at the context node c, the key constraint imposed by E above is as follows. For each i, $1 \leq i \leq n$, and for each c in $[\![t_1/\ldots/t_{i-1}]\!](root)$, let $S = [\![t_i]\!](c)$. Then, for each $s \in S$, there is set of bindings $v_i^1, \ldots, v_i^{k_i}$ for $\$'x_i^1, \ldots, \$'x_i^{k_i}$ such that

$$[\![t_i[p_i^1{=}\$'x_i^1, \ldots, p_i^{k_i}{=}\$'x_i^{k_i}]]\!](c) = \{s\}$$

That is, for each step in the path, the key bindings should exist and be unique. A key specified at a node which is not in $[\![t_1/\ldots/t_n]\!](root)$ is an error.

It can happen that the XML tag itself is an appropriate "key", therefore an extension of this syntax is required to bind variables to the tag names themselves e,g., $\ldots/t_{i-1}[\ldots]/\$'x_i \ldots$. The definition of key constraint is easily generalised. This constraint means that the children of node in $[\![t_1/\ldots/t_i]\!](root)$ have distinct tags. Also note that a consequence of our definition of a key constraint, a constraint of the form $/t_1 \ldots /t_{i-1}/t_i[\,]/\ldots/t_n$, in which the filter of the i^{th} step is empty means that any node in $[\![/t_1/\ldots/t_{i-1}]\!]$ has precisely one child with tag t_i.

[6]In XPath an empty filter as in /Root[ ] and /Data[ ] can be omitted. I have left it in to indicate the that it constrains the node to exist and to be unique.

[7] To be precise about the meaning of non-key bindings and constraints, we now consider expressions in which there are further non-key bindings for variables. Consider a constraint such as E above in which we have augmented the filter of the i^{th} step with an extra predicate of the form $q{=}\$^g y$:

$$/t_1[\ldots]/\ldots/t_i[p_i^1{=}\$'x_i^1,\ldots,p_i^{k_i}{=}\$'x_i^{k_i}, q{=}\$^g y]$$

in which q is a fully specified path, $\$y$ is a variable, and $\$^g$ is one four possible kinds of bindings, shortly to be specified.

We assume that the document satisfies the key constraint, therefore for each $c \in [\![t_1/\ldots/t_{i-1}]\!](root)$ and for each $s \in [\![t_i]\!](c)$ there is a unique set of bindings $v_i^1,\ldots,v_i^{k_i}$ for $\$x_i^1,\ldots,\$x_i^{k_i}$ such that

$$[\![t_i[p_i^1{=}\$v_i^1,\ldots,p_i^{k_i}{=}\$v_i^{k_i}]]\!](c) = \{s\}$$

Now consider the set V of distinct values for $\$y$ for which

$$[\![/t_1/\ldots/t_i[p_i^1{=}\$v_i^1,\ldots,p_i^{k_i}{=}\$v_i^{k_i}, q{=}\$y]]\!](c) = \{s\}$$

The meanings of the constraints imposed by the various bindings of the form $q{=}\$^g y$ are as follows:

- $q{=}\$.y$: $V = v$ (there is only one value) and y is bound to v.
- $q{=}\$?y$: $\mid V \mid \leq 1$ and if $V = \{v\}$, y is bound to v, otherwise y is bound to some null value.
- $q{=}\$^* y$: y is bound to V (no further constraints)
- $q{=}\$^+ y$: $\mid V \mid \geq 1$ and y is bound to V

Each such constraint is checked (and the bindings evaluated) independently.

[8]The constraints here are related to "strong keys", mentioned in [3] but not fully studied. Their precise definition is a bit subtle. We have chosen a definition that is local, in that it treats the variables at each step independently. This guarantees efficient checking, which can be done in linear time. Provided the total storage required for key data fits in main memory, constraint checking and citation generation can be performed by a two-pass traversal of large documents in secondary storage, and it may be possible to improve on this.

References

[1] Archival Resource Key. `http://www.cdlib.org/inside/diglib/ark/`. Retrieved on 10 Jan 2006.

[2] M. Benedikt, C. Y. Chan, W. Fan, R. Rastogi, S. Zheng, and A. Zhou. DTD-Directed Publishing with Attribute Translation Grammars. In *28th International Conference on Very Large Data Bases*, 2002.

[3] P. Buneman, S. Davidson, W. Fan, C. Hara, and W.-C. Tan. Keys for XML. *Computer Networks*, 39(5):473 – 487, August 2002.

[4] P. Buneman, S. Khanna, K. Tajima, and W.-C. Tan. Archiving Scientific Data. *ACM Transactions on Database Systems*, 27(1):2–42, 2004.

[5] The CIA World Factbook. `www.cia.gov/cia/publications/factbook/`. Retrieved on 8 Jan 2006.

[6] Consultative Committee for Space Data Systems. Reference Model for an Open Archival Information System. Technical Report CCSDS 650-B-1, National Aeronautics and Space Administration, Washington, DC 20546, USA, January 2002. Blue Book Issue 1.

[7] The Digital Object Identifier System. `http://www.doi.org/` . Retrieved on 10 Jan 2006.

[8] The Dublin Core Metadata. `http://dublincore.org/documents/2003/06/02/dces/`. Retrieved on 9 Jan, 2006.

[9] EMBL-EBI (European Bioinformations Institute). SPTr-XML Documentation. `http://www.ebi.ac.uk/swissprot/SP-ML/`. Retrieved in October 2001.

[10] W. Fan and L. Libkin. On XML Integrity Constraints in the Presence of DTDs. *Journal of the ACM*, 49(3):386–408, 2002.

[11] Excerpts from international standard iso 690-2 information and documentation – bibliographic references – part 2: Electronic documents or parts thereof. `http://www.collectionscanada.ca/iso/tc46sc9/standard/690-2e.htm#7.14`. Retrieved on 6 Feb, 2006.

[12] The official database of the IUPHAR Committee on Receptor Nomenclature and Drug Classification. `http://www.iuphar-db.org`. Retrieved on 8 Jan 2006.

[13] M. Lesk. *Practical Digital Libraries: Books, Bytes, and Bucks*. Series in Multimedia Information and Systems. Morgan Kaufmann, 1997.

[14] Online Mendelian Inheritance in Man, OMIM (TM). `http://www.ncbi.nlm.nih.gov/omim/`. Retrieved in October 2001.

[15] K. Patrias. National Library of Medicine Recommended Formats for Bibliographic Citation. . Supplement: Internet Formats. Technical report, National Library of Medicine,Reference Section Bethesda, MD 20894, July 2001. `http://www.nlm.nih.gov/pubs/formats/internet.pdf` Retrieved on 6 Feb, 2006.

[16] R. Snodgrass and C. Jensen. *Temporal Databases*. Morgan Kaufmann, March 2006.

[17] J. R. Walker and T. Taylor. *The Columbia Guide to Online Style*. Columbia, January 2001.

[18] XML Schema Part 0: Primer Second Edition. `http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/`, October 2004.